AI in Architectural Design. Adaptive Periodic Table for Classifying Cognitive Tiles and Hybrid Assemblies

Louis Roobaert[1,2(✉)] and Damien Claeys[1]

[1] TSA-LAB, LAB, Catholic University of Louvain, Brussels, Belgium
louis.roobaert@uclouvain.be
[2] Headlab of R&D Department, Senior Design Architect, B2Ai, Brussels, Belgium

Abstract. As digitalization reshapes all sectors, intelligent tools increasingly transform architectural design practices. Artificial Intelligence Systems (AIS) extend human cognition by processing large-scale data, performing complex calculations, and generating predictions. Yet, their adoption remains limited due to a lack of understanding, perceived complexity, and insufficient training.

This research introduces an adaptive periodic table to classify intelligent tools relevant to architectural design. Inspired by Mendeleev's model, it organizes learning algorithms according to three criteria: their computational complexity (eight periods, from constant to factorial), their learning methods (six families defined by supervised, unsupervised, or reinforcement learning, shallow or deep), and their role within AI processes (four blocks: analysis, model development, decision-making, causal inference). Methodologically, the table is constructed through a structured procedure that combines a transdisciplinary review of AI techniques, the definition of unified classification criteria, and the systematic mapping of algorithms that are already deployed or deployable in architectural workflows. Each algorithm is further specified through labels indicating types of inputs/outputs, accuracy, training time, and ability to capture linear or nonlinear relations.

By clarifying algorithmic functions and applications, the model provides a framework to compare and situate intelligent tools in design contexts. It fosters a better understanding of their interoperability and potential synergies, emphasizing that AI complements rather than replaces human cognition. This periodic table thus fulfils both pedagogical and professional functions: it supports architects and students in the use of AI by providing an intuitive understanding of its functionalities and integration, while serving as an operational reference that enables practitioners to select, combine, and sequence AI algorithms throughout the design process.

Keywords: Artificial Intelligence · Architectural Design · Periodic Table

A. Acevedo-De-los-Ríos et al. (Eds.): UES-EUS 2025, LNNS 1869, pp. 105–127, 2026.
https://doi.org/10.1007/978-3-032-19495-4_7

1 Introduction

As the digitalization of reality profoundly affects all sectors of human activity, the growing use of intelligent design-assistance tools is reshaping the practices of architectural design. Artificial Intelligence (AI) contributes to new hybrid configurations of reasoning, but these transformations remain largely unexplained from an epistemic perspective. Recent comparative studies between the cognitive spectra of humans and machines analyze the potential hybridizations within ecosystems of designers [1, 2].

While these transformations are evident at a technical level, they have not been accompanied by conceptual instruments allowing designers to understand how distinct AI techniques operate, differ, and can be meaningfully combined. However, both architectural education and professional practice provide limited access to the conceptual foundations of AI. As a result, designers often engage with intelligent tools without a clear understanding of their internal logics, limits, or combinatory potential.

Despite a growing body of literature on computational design, machine learning applications, and AI-assisted workflows, no existing study provides an accessible, systematic, and operational classification of learning algorithms specifically tailored to architectural practice. Existing resources remain either highly technical – addressed primarily to computer scientists—or focused on single applications without proposing a unifying epistemic framework. This absence of a coherent classificatory instrument limits designers' ability to understand, compare, and strategically combine algorithms within design processes.

More precisely, the objective of this research is to facilitate knowledge of intelligent design-assistance tools by proposing an adaptive tabular structure, allowing for their objective classification. Inspired by the periodic table of chemical elements, the proposed model aims to provide non-experts with an intuitive understanding of these tools potential and possible combinations, before turning them into partners for collaboration in design processes.

In this paper, AI and its subfields are first broadly defined through questions concerning data management and learning. A tabular classification model is then established (periods, families, blocks, labels). Selected tools relevant to architectural design are positioned within this structure. Finally, the relevance of the model is discussed through the introduction of a cognitive diptych – articulating tiles, assemblies, actors, and ecosystems – before concluding with its pedagogical and professional implications for architectural design.

2 Methods

The methodology involves developing a tabular model inspired by the periodic table of chemical elements, applied to artificial learning techniques relevant to architectural design. This section outlines the theoretical framework and the classification criteria that structure the model.

2.1 Theoretical Framework: Definitions and Structuring of AI

Given that intelligence has no definitive definition, artificial intelligence (AI) has been controversial since its introduction [3], and no universally accepted definition of AI or its subfields exists. Rather than attempting to define AI per se, it is more relevant to study the constitution of the discipline, its objectives, and the approaches used to pursue them.

As a data-driven problem-solving tool, AI aims to "emulate intelligent behaviors through a computer program without necessarily reproducing the corresponding functioning of the human being" [4]. This vast research program comprises a set of theories and procedures simulating human behavior to reproduce both cognitive reasoning and mechanical actions. Unlike classical algorithms dedicated to algebraic problem-solving (providing an exhaustive sequence of operations to solve a given problem), AI relies on heuristics, non-deterministic problem-solving methods based on trial and error, which do not guarantee outcomes but are generally faster.

Historically, two main approaches are distinguished: (1) the connectionist approach, leading to connectionism, modeling cognition as an emergent process and mimicking the physiological structure of the brain to create a universal neural machine, giving rise to the first artificial neural networks (NN) [5, 6]; and (2) the symbolic approach, leading to cognitivism, modeling mental processes through the symbolic functioning of human language and the manipulation of signs via logical rules, thereby establishing a universal symbolic machine, which underpinned the development of the first conversational agents and expert systems [7–9]. Today, the discipline has expanded considerably, notably due to the renewed enthusiasm for the connectionist approach (learning from experience), which had long been overshadowed by the successes of symbolic approaches (expert systems). Moreover, the two approaches are often hybridized—evolutionary algorithms testing variations, Bayesian inference incorporating uncertainty in learning, or analogy-based systems recognizing patterns across datasets [2].

As a discipline, AI has progressively emerged from the convergence of several fields [10] (see Fig. 1): (1) algorithmics (Alg); (2) computer science (CS), providing structured data, programming languages, and computational models [11]; (3) statistics (Sta), dealing with probability and the geometric structuring of data [12], including data science (DS) and big data (BD) management tools; (4) neurosciences (NS), which, through the connectionist approach (learning, synaptic plasticity, information processing), inspire various types of artificial neural networks (NN) [13]; and (5) cognitive sciences (SC), aligned with the symbolic approach, combining psychology, linguistics, and philosophy to study mental processes [14].

At the core of this transdisciplinary convergence, AI is here divided into two main categories: (1) machine learning (ML), emerging from symbolic approaches, understood as systems capable of making decisions from a dataset with a one-dimensional activation factor [15, 16]; and (2) deep learning (DL), grounded in connectionist approaches, understood as systems using multilayered artificial neural networks to model complex relationships between inputs and outputs, learning from large datasets, refining themselves through feedback loops, and performing multidimensional readings.

Although often considered an intermediate category between ML and DL, neural networks are more accurately situated within DL, here positioned at the intersection of NS and AI [17–19]. Many other subfields are clearly distinguished but are not addressed in this paper.

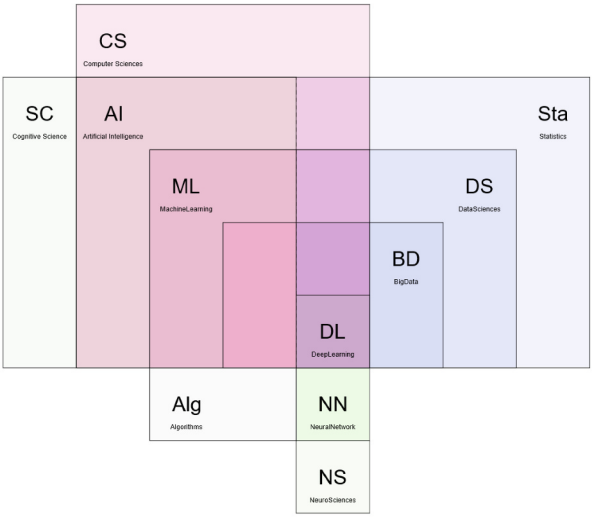


Fig. 1. The discipline of artificial intelligence and its main subfields.

The exponential growth of Artificial Intelligence Systems (AIS) [20] has prompted research groups to map this expanding landscape according to specific criteria. Among others, the Foundation Model Transparency Index (FMTI) was created by the Center for Research on Foundation Models (CRFM) at Stanford University, in collaboration with researchers from the MIT Media Lab and Princeton's Center for Information Technology

Policy [21]. The FMTI evaluates AIS transparency through a grid of indicators, requiring AI companies to submit detailed reports on multiple aspects of their models [22]. Another evaluative framework was proposed by François Chollet, an AI researcher at Google. Within the Open Ended Electronic Intelligent Robot Operating System (ONEIROS) project, he developed the open-source Python library Keras, designed to benchmark AIS performance by standardizing tests and enabling comparable performance measures across models. Keras models allow testing on standardized datasets (such as ImageNet) and measuring metrics such as accuracy, inference speed, and computational efficiency [23].

In complement to these evaluative studies analyzing and comparing AIS increasingly hybridizing our environments, this paper proposes a different approach. Rather than focusing on AIS themselves, the objective here is to establish a cartography of distinct techniques at the algorithmic level of AI—before they are embedded into systems. The idea is to construct a periodic table of artificial learning algorithms relevant to architectural design, whose combinatory use can generate AIS.

2.2 Operational Framework: Construction of the Tabular Model

A structure is proposed to build a table allowing the classification of different intelligent tools used in architectural design. Originally attributed to the Russian chemist Dmitri Ivanovich Mendeleev [24], the periodic table of elements has been adapted many times before taking on its current form. The known chemical elements are arranged according to their physical and chemical properties, while leaving room for the later inclusion of yet unknown elements. The present table is structured around three main organizational principles: seven horizontal periods (horizontal alignments of elements sharing the same level of electronic energy), eighteen vertical families (vertical groupings of elements having the same number of valence electrons), and four blocks (clusters of elements whose valence electrons belong to the same electronic sub-shells).

By analogy, the periodic table of artificial learning adopts a similar configuration through four steps:

(1) The periods (horizontal axis): An eight-level vertical scale of algorithmic complexity is established, allowing the tools to be ordered into eight corresponding horizontal periods.
(2) The families (vertical axis): The tools are then sorted into six vertical families according to two criteria: (a) their learning method (supervised, unsupervised, reinforcement), and (b) their depth (shallow or deep). Each family groups techniques sharing a common learning logic.
(3) Blocks (transversal organization): Four transversal blocks are defined, reproducing the succession of the main phases of an AI workflow: analysis, modelling, decision-making, and inference. These blocks clarify the functional role of each algorithm within data-processing pipelines.
(4) Label (identification): Each algorithmic tool receives an individual label indicating its name, primary function, and possible variants, forming the elementary unit of the table.

The following four sections detail the structural components of the table: periods, families, blocks, and labels.

Levels of Algorithmic Complexity (Periods)
At first glance, intelligent tools may be distinguished by their material limits (e.g., types of RAM or processors used) or software constraints (e.g., programming languages). However, their effectiveness is classified here formally, according to the general value of their level of algorithmic complexity, calculated independently of their execution environment [25].

In theoretical computer science, the degree of complexity of a problem corresponds to the level of difficulty required for its resolution. Complexity is both spatial (memory space, or the size occupied by the data involved in the resolution process) and temporal (execution time, or the number of elementary operations required to complete the process). A complexity class is therefore a set of algorithmic problems whose resolution requires the same quantity of resources. Depending on the complexity degree to which an algorithm responds, it can be distributed into a class of efficiency, each rigorously defined by asymptotic growth orders (upper bounds of mathematical functions).

Introduced in the late nineteenth century by number theory [26] before being widely adopted in computer science to analyze algorithmic complexity [27, 28], Big-O notation designates the order of an algorithm's complexity, while the variable n expresses data-related complexity. It provides a standardized framework for evaluating the efficiency of algorithms in terms of temporal or spatial complexity, without implementing them or measuring empirical performance.

In the table, intelligent tools are classified into eight horizontal periods, each corresponding to a level of complexity, arranged vertically from least to most complex: (1) constant $O(1)$; (2) logarithmic $O(\log n)$; (3) linear $O(n)$; (4) linearithmic $O(n \log n)$; (5) quadratic $O(n2)$; (6) cubic $O(n3)$; (7) exponential $O(2^n)$; (8) factorial $O(n!)$. Thus, the complexity of an algorithm noted $O(1)$ does not vary with dataset size, while that of an algorithm noted $O(n2)$ grows quadratically (doubling the data size quadruples execution time). By plotting dataset size (n) on the x-axis and processing time (t) on the y-axis, algorithms can be compared graphically: the steeper the slope, the higher the algorithmic complexity [29].

Learning Methods (Families)
In the table, intelligent tools are classified according to the type of learning method they employ, arranged by columns into six basic families. Two classification criteria are used to define these six families: (1) the learning mode of the tool ((I) supervised, (II) unsupervised, (III) reinforcement), spanning the continuum between human intervention and autonomy; and (2) the depth of learning ((a) shallow, (b) deep). Since the last three involve DL (deep learning), they are more complex versions of the first three involving ML (machine learning). The table is therefore subdivided into six column types:

(1) Supervised shallow learning (I.a – N): groups algorithms trained on labeled datasets (classified with known attributes and associated with known responses) to establish correspondences between inputs and outputs. This method is highly precise but requires human preprocessing of the data and has limited generalization capacity.

Commonly used techniques include classification and regression, linear regression, or supervised neural networks [23].

(2) Unsupervised shallow learning (II.a – S): encompasses algorithms capable of identifying patterns and structures in datasets without explicit guidance. Starting from raw data, the algorithm learns to detect models, structures, and implicit relationships between inputs and outputs. This approach is significantly more complex, as it requires the algorithm to develop its own interpretation of the data without explicit directives. Frequently used techniques include clustering (grouping by similarity), association, or dimensionality reduction [30].
(3) Shallow reinforcement learning (III.a – R): refers to autonomous algorithms that must learn by themselves which actions to undertake. This type of learning allows artificial agents to acquire knowledge from their own experience, without being provided with data, through a cumulative system of rewards and penalties [31].
(4) Supervised deep learning (I.b – NP): describes neural networks trained on labeled datasets, capable of predicting outputs with greater accuracy from given inputs, as in speech recognition or natural language processing[32].
(5) Unsupervised deep learning (II.b – SP): encompasses structures such as GANs or autoencoders, used to discover structures within unlabeled data. These methods enable the learning of multiple levels of features or representations, making them particularly useful for generative tasks such as producing images or text [33].
(6) Deep reinforcement learning (III.b – RP): combines the principles of deep learning with reinforcement learning algorithms, enabling an autonomous agent to make strategic decisions and optimize its behavior to maximize the rewards it receives [34].

Phases of Intelligent Processes (Blocks)

Problem-solving with intelligent tools passes through four fundamental phases linked to data management and learning. Based on the six learning families, four blocks are defined:

(1) Data analysis (input processing and collection): the designer gathers and analyzes the data at their disposal, whether drawn from past experiences or collected specifically for the design process at hand. By mobilizing cognitive skills, the data are processed and structured in ways that enable visualization or comparison. This processing phase is polymorphic, ranging from clustering to filtering and classification, thereby organizing data in a structured manner that facilitates pattern recognition and the deduction of insights.
(2) Model development (building and adjusting): akin to the process of shaping data into multiple admissible design proposals, the analyzed data and resulting insights are used to develop learning models capable of producing predictions. Predictive models operate by leveraging characteristics of past data to generate forecasts.
(3) Decision-making (based on model results): comparable to evaluating different design alternatives to select one, decision-making balances exploration—testing new alternatives with their attendant risks and costs—against exploitation, which relies on proven strategies and successful procedures from the past [31].

(4) Causal inference (verification and interpretation): analogous to verify and assess the chosen design proposal, where the designer runs simulations to confirm that the project meets expectations and satisfies predefined criteria. Models developed with AI are rigorously tested to validate their precision, recall, generalizability, and robustness. In particular, causal inference identifies the causes of a phenomenon and relates them to the observed effects, thereby providing feedback on the validity of decisions taken. At each stage, deep learning may implement a variety of neural architectures, using activation functions to regulate neuron activation. Optimization and regularization methods are further applied to refine models and prevent overfitting [35].

In practice, these four stages form the basis of a data scientist's work when establishing an algorithmic system to solve a given problem. By analogy, a partial parallel can be drawn with the phases of architectural design. Architectural design may be considered as an attempt to resolve a particular class of ill-defined problems [36], which require specific resolution strategies. Rather than being linear, such processes are circular, iterative, and reflexive, alternating between phases of convergence and divergence before arriving at a suboptimal solution [36, 37]. In this context, the designer operates through a dynamic combination of intelligent design-assistance tools, constantly navigating between them to refine the project under development.

Properties of Intelligent Tools (Labels)

To improve the understanding of the potential effects of intelligent tools in architectural design, each algorithm is precisely identified by a label, in addition to its algorithmic complexity, learning method, and process phase. This label comprises five complementary pieces of information (see Fig. 2):

(1) Type of inputs: describes the nature of inputs processed by the algorithm—numerical (N), audio (A), video (V), or text (E). Some algorithms use multiple types [38].
(2) Type of outputs: describes the nature of the algorithm's results, which may be continuous (production of a value belonging to the same class as the input), categorical (classification of inputs), sequential (series of values or labels), structured (organization of inputs into a new label), ranked (ordering inputs according to specific metrics), or probabilistic (measuring uncertainty) [39, 40].
(3) Accuracy (positive predictive value): indicates the proportion of correct predictions (true positives) relative to the total number of positive predictions (sum of true positives and false positives). Accuracy is represented on a scale from 1* (minimal) to 4* (maximal).
(4) Training time: indicates the speed at which the algorithm learns from training data before reaching an acceptable threshold of performance or predictive accuracy. It is represented on a scale from 1* (slow) to 4* (very fast) [41].
(5) Linearity: specifies whether the algorithm models linear or non-linear relationships that is, whether every change in the inputs is directly proportional
(6) to changes in the outputs. Some algorithms are capable of modeling both types of relationships simultaneously [41].

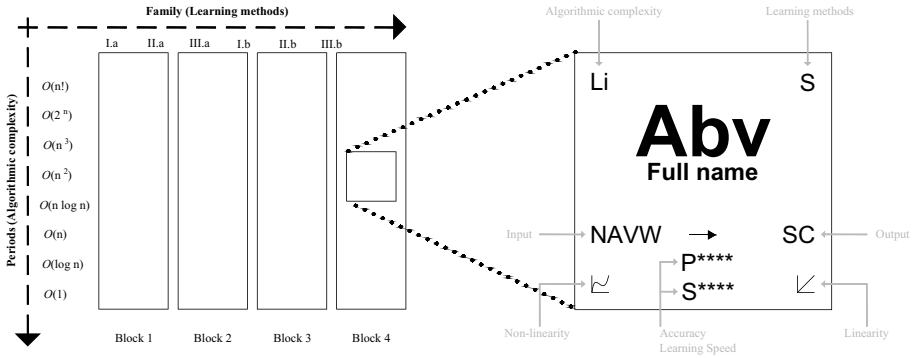


Fig. 2. (a) Structure of the table through periods, families, and blocks. (b) Structure of a label.

3 Results: Placement of Algorithmic Varieties in the Table

After establishing the general structure of the tabular model of intelligent tools (periods, families, blocks, labels), its relevance is tested through a non-exhaustive positioning of algorithmic varieties. The adaptive character of the tabular model incorporates the possibility of modifying and adding algorithms, thereby reflecting the dynamic and cyclical nature of design processes as well as the continuous introduction of new intelligent tools. Its adaptability is validated by applying the same classificatory rules to additional algorithms identified in the literature; when these new elements can be integrated into existing cells or into intentionally reserved empty positions—without altering the underlying organizational principles of periods, families, blocks, and labels—the model is deemed to preserve its conceptual stability while remaining extensible.

The selection of algorithms included in the table is based on four criteria: (a) only those documented in the scientific literature as contributing to computational design are retained; (b) priority is given to algorithms that are widely implemented across computational ecosystems; (c) the selected algorithms must possess a clearly identifiable learning logic and represent a distinct computational principle; (d) they must illustrate the functioning of different learning processes and exhibit combinatory potential.

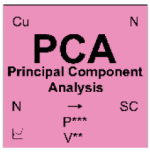
Before examining each block in detail, an overview of the algorithms positioned within the table is provided. Each algorithmic family is represented by a set of techniques that illustrate the main computational principles associated with the four blocks:

- Block 1 – Data Analysis: Principal Component Analysis (PCA), Clustering (Clu), K-Means (KM), Topic Model (TM), Latent Dirichlet Allocation (LDA), Support Vector Machine (SVM), Rules Learning (RuL).
- Block 2 – Model development: Regression (Reg), Linear Regression (LR), Ridge Regression (RR), Logistic Regression (LoR), Classification (Cla), Quadratic Discriminant Analysis (QDA), Simple Neural Network (NN), Complex Neural Network (CNN), Biological Networks (Bio), Generative Adversarial Network (GAN), Generative Modeling (GM), Transformer (Tra), Generative Pretrained Transformer (GPT), Variational Autoencoders (VAE), Dropout (Dro), Max Pooling (MP).
- Block 3 – Decision making: Model Predictive Control (MPC), Random Forest (RD), Multi-Armed Bandit (MAB), Lasso Regression (Las), Markov Decision Process (MDP), Dynamic Bayesian Networks (DBN), Actor-Critique (AC), Q-Learning (QL), Epsilon-Greedy (EG).
- Block 4 – Causal inference: Causal Forest (CF), Instrumental Variable Regression (IVR), Causal Decision Tree (CDT).

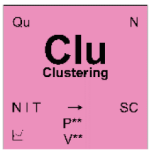
The algorithms presented below are described block by block. Each algorithm is represented by a tile that shows its position within the tabular model by indicating the block, family, and period to which it belongs, as well as its labels (input types, output types, accuracy, training time, linearity).

3.1 Data Analysis (Block 1)

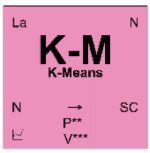
This block contains all algorithms dedicated to processing data and facilitating its visualization. Three main categories of algorithms can be identified, each comprising a diversity of sub-algorithms [42]: (1) dimensionality reduction, which enables the treatment of high-dimensional data, where the number of variables, labels, and attributes prevents meaningful correlation for human cognition [43]; (2) clustering, which groups data into homogeneous subsets, revealing internal structures and patterns; (3) classification, which assigns labels to unlabeled data based on learned models [44]. The selection of a given class of algorithms depends on the type and structure of the data available to the designer, while the data type (text, numerical, video, image) determines the most appropriate tool for analysis [17].



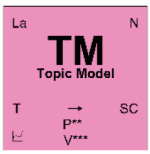
Principal Component Analysis (PCA): reduces the dimensionality of a dataset, simplifying complex data into a smaller number of uncorrelated variables.



Clustering (Clu): explores a dataset and groups entries according to similarities, highlighting recurring characteristics and patterns while simplifying spatial visualization of the inputs.

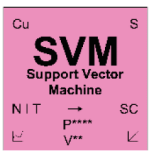


K-Means (KM): a sub-technique of clustering that identifies natural clusters by minimizing intracluster variance, based on the distance of each data point to the nearest mean [45].

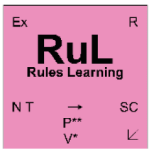


Topic Model (TM): identifies recurring themes in textual inputs. It facilitates subject retrieval, classifying and structuring information by performing thematic analysis to detect predominant topics and associated keywords.

Latent Dirichlet Allocation (LDA): similar in concept to K-Means but specific to textual analysis, it helps identify trends in a document corpus.



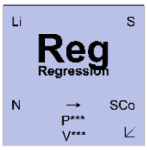
Support Vector Machine (SVM): analyzes the way high-dimensional data is distributed on a Cartesian hyperplane, separating it into classes based on the distance to the closest data points of each class, referred to as the maximum margin [46].



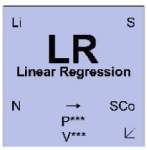
Rules Learning (RuL): rule-based learning analyzes a dataset and distinguishes between entries that respect or violate predefined rules (such as urban regulations or safety standards).

3.2 Model Development (Block 2)

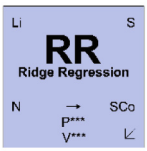
This block contains networks and models that enable the development and adjustment of predictive systems. The purpose of these elements is to support decision-making based on processed data. This section usually contains three main families: (1) regression, comprising methods that estimate relationships between variables and predict continuous values [47]; (2) classification, encompassing techniques that assign discrete categories to data instances; (3) neural networks, which model complex nonlinear interactions between inputs and outputs, simulating aspects of the human brain to recognize intricate patterns [48].



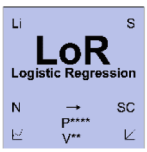
Regression (Reg): regression identifies causal relationships within historical datasets and builds models to make predictions. It helps assess the impact of variables on quantitative outcomes (Freedman, 2009).



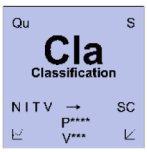
Linear Regression (LR): considered the most elementary regression model, it establishes linear relationships between independent and dependent variables, using weights to evaluate the contribution of each attribute to a target variable.



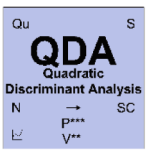
Ridge Regression (RR): extends LR by adding a penalty term to reduce model complexity, thus helping to prevent overfitting and improve prediction accuracy.



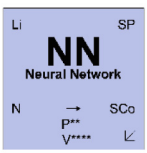
Logistic Regression (LoR): an adaptation of LR for classification problems, particularly binary responses (true/false). Widely used to model event probabilities based on predictors, it identifies critical success factors.



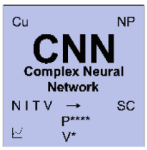
Classification (Cla): classification organizes data into discrete categories. It identifies patterns and relationships in a dataset to categorize new entries based on predefined classes [49].



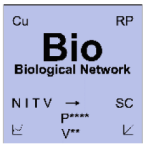
Quadratic Discriminant Analysis (QDA): a more sophisticated version of classification, QDA uses quadratic functions to provide curved decision boundaries, producing more refined predictive models than the linear version.



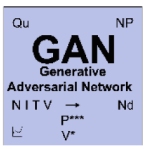
Simple Neural Network (NN): inspired by the early work of McCulloch and Pitts, simple neural networks consist of a single layer connecting inputs to outputs. Inputs (independent variables) are weighted and combined with a constant to produce outputs (dependent variables) through an activation function [5].



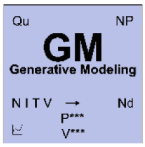
Complex Neural Network (CNN): more advanced than simple NN, CNNs include hidden layers between input and output, performing intermediate calculations until a final result is reached. Their architecture depends on the number of hidden layers, the number of neurons per layer, and the typology of connections between neurons.



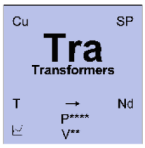
Biological Networks (Bio): CNNs inspired by the architecture and functioning of biological neural systems. They attempt to model brain mechanisms to create algorithms capable of learning and processing information in ways analogous to living organisms.



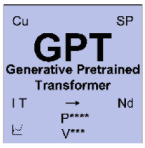
Generative Adversarial Network (GAN): an unsupervised learning model comprising two neural networks—a generator and a discriminator—trained simultaneously in a competitive game. The generator produces data resembling the input set, while the discriminator attempts to distinguish generated from real data [50].



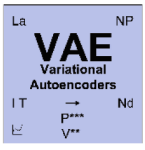
Generative Modeling (GM): learns to generate new data similar to the training set by modeling statistical distributions. Applied in text synthesis, image creation, and other generative tasks.



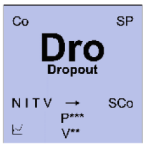
Transformer (Tra): a neural architecture specific to natural language processing, relying on attention mechanisms to improve output quality. Unlike CNNs, Transformers model dependencies across entire sequences.



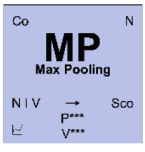
Generative Pretrained Transformer (GPT): a natural language processing model based on the Transformer architecture. Pretrained on large text corpora, it can be fine-tuned for a wide range of linguistic tasks, from speech recognition to summarization and question answering.



Variational Autoencoders (VAE): produce new data by transforming input data into a compressed latent space and reconstructing it into another form [50].



Dropout (Dro): a regularization method used in CNNs. It randomly ignores a selection of neurons during gradient calculation, simplifying computations, reducing complexity, and lowering overfitting risk [51].



Max Pooling (MP): commonly used in convolutional neural networks, this technique simplifies information by selecting the maximum value within each subregion of the previous layer, reducing dimensionality while preserving robustness to small variations in scale [52].

3.3 Decision-Making (Block 3)

Decision-making largely depends on the nature and quantity of available data. After an autonomous learning phase, AI models can generate predictions (phase 2). Based on these predictions, a decision can be made depending on two factors: (1) the availability of data (from low to high), and (2) the state of the environment (fixed or dynamic). The environment is considered fixed when it does not change, or changes very slowly (e.g., the existing built environment), and dynamic when it evolves with every action, interaction, or time step (e.g., during the design of an architectural project). By crossing data availability with environmental state, four distinct decision-making scenarios can be defined, each influencing the choice of intelligent tools.

Model Predictive Control (MPC): an algorithmic model aimed at improving future decisions. It uses historical data to learn the state of an environment, then predicts and optimizes the output trajectory [53].

Random Forest (RD): an ensemble of decision trees, each built from a sample of data and using a random subset of variables at each split. It improves decision-making and reduces overfitting. Each tree votes for a class, and the majority is selected as the final prediction.

Multi-Armed Bandit (MAB): balances exploration and exploitation across a set of alternative options. The algorithm evaluates all possibilities and selects the one offering the highest reward. Its name derives from the analogy with a gambler playing multiple slot machines simultaneously.

Lasso Regression (Las): a technique for improving model predictability and interpretability by introducing a regularization penalty on coefficients, encouraging automatic variable selection. Widely used for datasets with a large number of variables [54].

Markov Decision Process (MDP): a model simulating constant interaction between an agent and an environment. The agent chooses actions, and the environment reacts by presenting new situations. When multiple decisions are at stake, the agent's ultimate goal is to maximize cumulative rewards over the course of the process.

Dynamic Bayesian Networks (DBN): model data sequences or processes evolving over time, recognizing patterns through a set of interdependencies among variables.

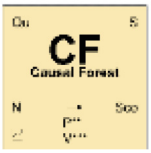
Actor-Critique (AC): combines two main components: the actor, which selects actions, and the critic, which evaluates those actions by computing a value function. This dual process reduces the variance of reward estimations [31].

Q-Learning (QL): enables an agent to learn a value function without modeling the environment, estimating the optimal value of actions possible from a given state. As decisions are made, QL updates its values using obtained rewards while maximizing expected future rewards.

Epsilon-Greedy (EG): a strategy that balances exploration and exploitation. At each step, the agent chooses a random action (exploration), then selects the action with the highest estimated value (exploitation) based on probability [55, 56].

3.4 Causal Inference (Block 4)

Causal inference focuses on identifying and attributing causes to observed phenomena. Its objective is to confirm the existence of cause–effect relationships between events, thereby enabling the evaluation of the effectiveness of decisions made (Fig. 3).



Causal Forest (CF): estimates causal effects on an outcome variable by modeling the relationship between treatment variables, covariates, and results. By generating multiple decision trees on random subsets of data, it deduces how changes in one variable influence another.

Instrumental Variable Regression (IVR): estimates causal relationships when a regression model is subject to endogeneity, i.e., when explanatory variables are correlated with the error term. IVR provides unbiased estimates of causal effects by introducing instrumental variables.

Causal Decision Tree (CDT): identifies and quantifies the causal effects of an intervention on an outcome by subdividing data into homogeneous subgroups based on covariates. Each node of the tree represents a decision based on a variable, and the leaves provide an estimate of the causal effect of the intervention for individuals in the corresponding subgroup [57].

Among all the algorithms identified, only those presented here have been integrated into the periodic structure, with the aim of demonstrating its conceptual coherence and methodological validity. The illustration above therefore represents a provisional state,

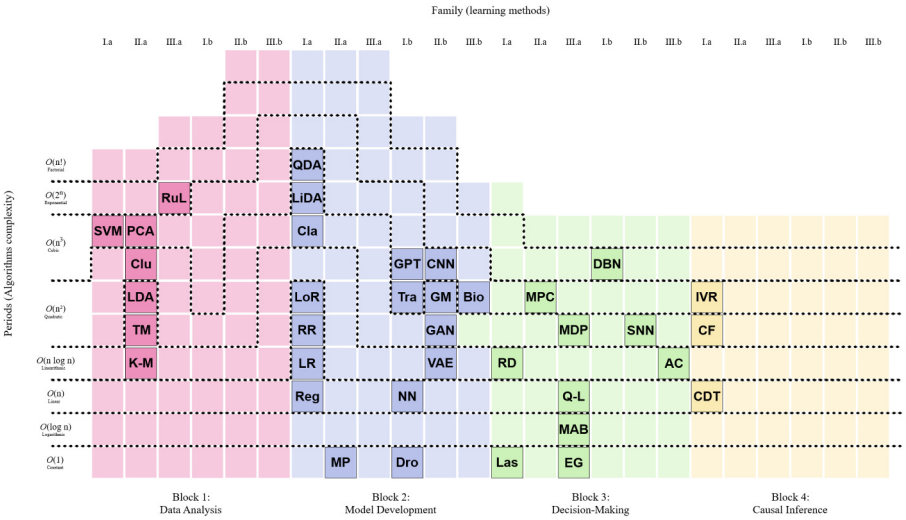


Fig. 3. The adaptive periodic table of intelligent design-assistance tools.

corresponding to a given moment t, of the algorithms currently listed among the existing approaches.

To enable the table to accommodate and classify additional algorithms in the future, the proposed tabular form has been designed as an open and evolving structure: for each possible combination of block, family, and period, an empty cell has been intentionally left free, allowing the insertion of new elements as the field develops. The dotted lines, meanwhile, are deliberately non-rectilinear: they delineate the periods, acknowledging that several algorithms may share the same class of computational complexity, which necessitates their stacking within the table. Finally, the white vertical separation lines mark the boundaries between the different learning families, ensuring a structured and legible organization of the overall tabular system.

Applying the tabular model to the selected algorithms reveals configurations that shed light on how intelligent techniques are currently mobilized in architectural design. First, a concentration of techniques appears within the intermediate complexity periods (O(n) to O(n log n)). This distribution shows that architectural workflows tend to favour algorithms offering a compromise between computational cost, interpretability, and operational robustness. High-complexity methods—exponential or factorial—are present but comparatively rare, due to computational limitations.

Second, the distribution of techniques across the four blocks is uneven. Blocks 1 (data analysis) and 2 (model development) contain most of techniques positioned within the table, indicating that AI-assisted processes prioritize exploratory data processing and predictive modelling rather than decision-making (Block 3) or causal reasoning (Block 4). This imbalance highlights a disciplinary bias: designers primarily use AI to analyze datasets or generate design alternatives, while algorithmic tools capable of supporting long-term planning, optimization under uncertainty, or causal verification remain underutilized.

Third, the distribution of learning families shows a predominance of supervised techniques, whether shallow or deep, consistent with the prevalence of labelled datasets. Unsupervised and reinforcement-learning techniques are represented but far less frequent. Their relative scarcity reveals the limited penetration, in current design environments, of autonomous or self-organizing models—despite their potential for exploratory search, multi-agent coordination, or adaptive design strategies.

Fourth, the deep learning techniques included in the model cluster almost exclusively within the highest complexity periods (from O(n3) upwards). Their placement demonstrates both their computational demands and their capacity to encode high-dimensional, nonlinear relationships—properties that make them particularly suitable for generative modelling, environmental simulation, or multimodal architectural tasks. This concentration reinforces the structural distinction between machine-learning-based and deep-learning-based techniques already embedded in the architecture of the table.

Finally, the presence of intentionally empty cells at the intersection of several periods, families, and blocks highlights underexplored areas of the design space. These absences point to methodological opportunities: entire classes of algorithms not yet mobilized in design workflows could, in principle, be integrated into the periodic structure. Identifying these vacant positions constitutes, in itself, a productive research outcome, as it enables both researchers and practitioners to anticipate future extensions of the field and to identify algorithmic techniques that may be meaningfully adapted to architectural design tasks.

4 Discussion: Cognitive Diptych – Tiles, Assemblies, Actors, Ecosystems

In continuity with systemic approaches initiated by von Bertalanffy (1968), Miller (1978), Forrester (1961), and Le Moigne (1990) [58–61], cognition can be understood as an open system, characterized by information exchanges, feedback loops, and multi-level organization. A system cannot be reduced to the sum of its parts but emerges from the dynamic articulation of its components, their interrelations, and their embedding within an environment [62]. This perspective implies that cognitive processes should not be considered in isolation but rather in their co-organization and hybridization modes. Accordingly, the proposed tabular model does not only aim to classify artificial cognitive operations but also to make intelligible the relations between cognitive units, their combinations, and their integration within dynamic ecosystems.

Applied here specifically to artificial forms of cognition, the table is part of a broader research trajectory seeking to construct a symmetric cartography of cognitive functions, whether human or artificial [2]. Just as algorithms can be decomposed into cognitive tiles, human mental processes may also be described as sets of functions, forming all possible combinations of projective activities engaged in facing ill-defined and evolving problems [63–65]. These functions, following their own internal logic, can also be organized into a periodic tabular structure.

Placing these tables in symmetry highlights that human and artificial tiles do not function hermetically, but rather overlap, extend, or transform one another. For example, biological memory and artificial memory do not oppose but hybridize within instrumented

devices [66, 67]. Similarly, the abductive reasoning of the designer can be supported or amplified by generative algorithms [50, 68].

From this perspective, the artificial table presented here must be understood as half of a cognitive diptych: on one side, the structuring of human functions; on the other, the classification of computational capacities. Their continuity establishes a space of operational correspondence, enabling the identification of different regimes of coupling.

This theoretical structure can be translated into an operational methodology for hybridizing cognition within the design ecosystem.

Cognitive tiles constitute the smallest unit of cognition: on the natural side, they refer to elementary mental functions; on the artificial side, they correspond to algorithms. Each tile possesses a limited yet precise scope of action, contributing to a specific operation within the cognitive chain. The table thus functions as a reference system that designers can mobilize to identify the algorithmic operations required at different phases of the process: data extraction and structuring (block 1), model production (block 2), strategic decision-making (block 3), or causal verification (block 4). Since human cognitive architecture rarely mobilizes all operations simultaneously, this decomposition helps designers articulate their own capacities with those of the tools by selecting the artificial tiles best suited to complement or extend their mental operations.

At a higher level, tiles combine to form cognitive assemblages—operative elements that express cognitive syntax and may become hybrid when human and artificial tiles are associated within the same process [69–71]. This correspondence enables the structuring of chains of hybrid tools (for example, combining a clustering technique, a generative model, and a multi-criteria evaluation tool) adapted to the resolution of a given problem. These assemblages can then be sequenced, adjusted, or retro-activated according to a workflow logic incorporating feedback loops.

At the next level, assemblages are embedded within concrete systems that carry and give them operational materiality: plugins, scripts, modules, pipelines, computational environments, or interfaces. These carriers temporarily stabilize cognitive assemblages, facilitate their reuse, and enable the circulation of information, matter, and energy within a workflow. They therefore act as mediators that make assemblages truly actionable in practice, orchestrating the articulation between human and computational operations.

Above the actors lie cognitive ecosystems—distributed environments in which a plurality of human, artificial, and hybrid actors interact. These ecosystems constitute the true ecology of design [72, 73], within which cognitive assemblages circulate, recombine, and regulate one another. Far from being static, they embody dynamic environments where individual intelligences are continuously reconfigured through their collective inscriptions, generating new forms of cooperation and shared creativity.

Although the model provides a structure for classifying cognitive operations, it presents several limitations. First, it depends closely on the current state of the literature and on the selection of algorithms documented in the field of computational design; as a result, some techniques relevant in other domains may be absent. Second, the table does not evaluate the actual performance of algorithms in design situations: it offers a conceptual and operational cartography, but its empirical relevance depends on workflows, datasets, and computational environments specific to each context. Finally,

the operational validity of the model has not yet been demonstrated; it will need to be assessed through experiments and case studies that will extend this research (Fig. 4).

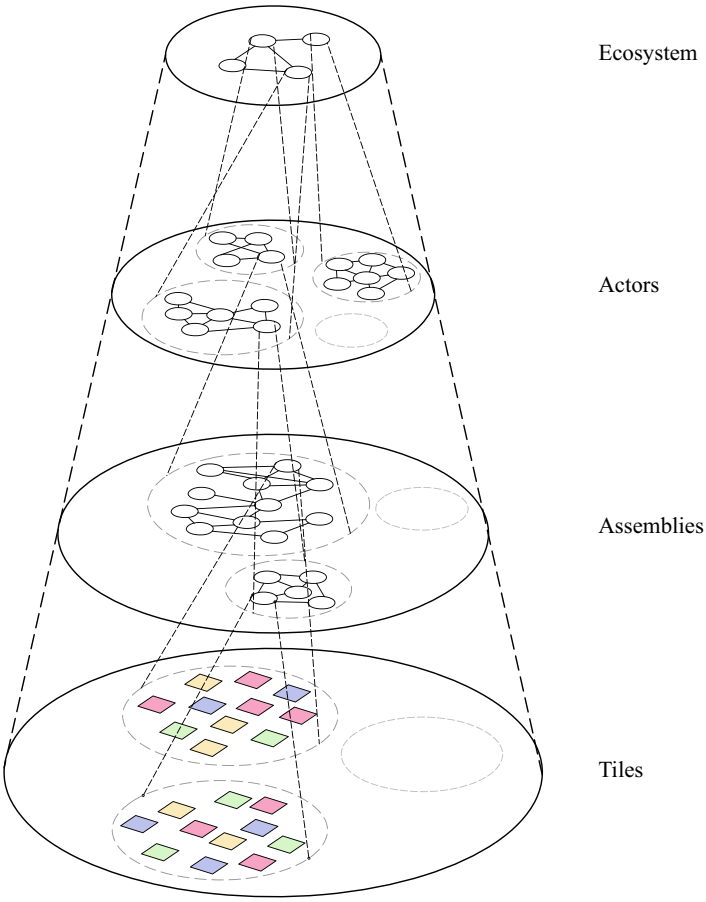


Fig. 4. Cognitive Diptych – Tiles, Assemblies, Actors, Ecosystems

5 Conclusions

In the same way that Mendeleev's periodic table classifies chemical elements and anticipates their potential combinations into molecules, or that visual scripting software enables combinations of intelligent components to generate scripts capable of solving specific problems (e.g., Grasshopper or Dynamo), the adaptive and evolutionary periodic table of design-assistance tools presented here aims to help non-expert designers situate and anticipate the interoperability of the various tools available by visually comparing the algorithms they mobilize.

This combinatory approach serves at least five purposes: (1) partially demystify the AI processes that can be mobilized in architectural design; (2) clarify the individual functions of each algorithm, as well as their inputs and outputs; (3) identify potential synergies among classified algorithms to address complex problems; (4) support anticipation and decision-making, by imagining processes that associate several tools prior to implementation and by identifying solutions best adapted to computational power and data availability; (5) anticipate interoperability across all phases of a design process using AI.

The positioning of intelligent tools within the proposed table demonstrates the relevance of its structure. The goal was not to definitively classify every existing algorithm (only those with direct relevance to architectural design were prioritized), but rather to provide a reference model situating each algorithm in relation to others within a coherent structure.

Reflection on a tabular model for classifying intelligent tools contributes to deepening the understanding of the cognitive spectrum within hybrid design networks—encompassing humans, machines, and cyborgs. This research introduces a framework for apprehending the interrelations between AI and the architect's tools. Beyond its classificatory function, the adaptive table constitutes a methodological contribution by offering a systematic framework through which designers can analyze, compare, and operationally combine heterogeneous AI techniques within coherent workflows. The adaptive nature of the table also opens concrete applications for architectural education, where it can serve as a structured learning scaffold; for professional practice, where it can support the design of AI-augmented workflows; and for future developments, by providing a stable framework into which new algorithms and emerging learning paradigms can be progressively integrated. It functions simultaneously as a pedagogical instrument, decoding and rendering accessible the concepts of machine learning and deep learning, and as a professional instrument, enabling their integration both in practice and in architectural education.

References

1. Claeys, D.: Entre cogitation et computation : Comment déjouer l'obsolescence programmée de l'architecte ? In: Anticrise architecturale : analyse d'une discipline immergée dans un monde numérique, pp. 157–176. Louvain-La-Neuve (2021)
2. Roobaert, L., Claeys, D., Cleven, S.: Espaces d'hybridation en conception architecturale : modalités collaboratives entre cognitions naturelles et artificielles. In: SHS Web of Conferences. Tunis, Tunisia (2024). https://doi.org/10.1051/shsconf/202420304002
3. McCarthy, J., Minsky, M.L., Rochester, N., Shannon, C.E.: A proposal for the Dartmouth summer research project on artificial intelligence. Artif. Intell., 1–12 (1955). https://doi.org/10.1609/aimag.v27i4.1904
4. Haton, J.-P., Haton, M.-C.: L'Intelligence artificielle. Presses universitaires de France, Paris (1989)
5. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biol. **65**(5), 99–115 (1943). https://doi.org/10.1016/S0092-8240(05)80006-0
6. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**(6), 386–408 (1958). https://doi.org/10.1037/h0042519

7. Turing, A.: Computing Machinery and Intelligence. Mind. LIX(236), pp. 433–460 (1950). https://doi.org/10.1093/mind/LIX.236.433
8. Von Neumann, J.: First draft of a report on the EDVAC. Moore Sch. Electr. Eng., Univ. Pennsylvania **15**(4), 27–72 (1945)
9. Weizenbaum, J.: ELIZA—a computer program for the study of natural language communication between man and machine. IEEE Annn. Hist. Comput. **15**(1), 36–45 (1966). https://doi.org/10.1145/365153.365168
10. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson, Upper Saddle River (1995)
11. Brooks, R.A.: Intelligence without representation. Artif. Intell. **47**(1), 139–159 (1991). https://doi.org/10.1016/0004-3702(91)90053-M
12. Efron, B., Hastie, T.: Computer Age Statistical Inference: Algorithms, Evidence, and Data Science. Cambridge University Press, New York (2016)
13. Kandel, E.R., Schwartz, J.-H., Jessel, T.M., Siegelbaum, S., Hudspeth, A.J.: Principles of Neural Science. McGraw-Hill, New York (2013)
14. Gazzaniga, M.S.: The Cognitive Neurosciences. MIT Press, Cambridge (2018)
15. Samuel, A.L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. **3**(3), 210–229 (1959). https://doi.org/10.1147/rd.33.0210
16. Bishop, C.M.: Training with noise is equivalent to Tikhonov regularization. Neural Comput. **7**(1), 108–116 (1995). https://doi.org/10.1162/neco.1995.7.1.108
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
18. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. **79**(8), 2554–2558 (1982). https://doi.org/10.1073/pnas.79.8.2554
19. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**(1), 85–117 (2015). https://doi.org/10.1016/j.neunet.2014.09.003
20. Silva, C.S.R., Fonseca, J.M.: Artificial intelligence and algorithms in intelligent systems: advanced analytics: moving forward artificial intelligence (AI), algorithm intelligent systems (AIS) and general impressions from the field. In: Silhavy, R. (ed.) Artificial Intelligence and Algorithms in Intelligent Systems, pp. 308–317. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-319-91189-2_30
21. Bauer, W., Vocke, C.: Work in the age of artificial intelligence – challenges and potentials for the design of new forms of human-machine interaction. In: Kantola, J.I. and Nazir, S. (ed.) Advances in Human Factors, Business Management and Leadership, pp. 493–501. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-20154-8_45
22. Bommasani, R., et al.: The foundation model transparency index (2023)
23. Chollet, F.: Deep learning with Python. Manning Publications, Shelter Island (2021)
24. Dmitri, M.: La dépendance entre les propriétés des masses atomiques des éléments. Presentation to the Russian Chemical Society, Russia (1869)
25. Gustafson, J., Rover, D., Elbert, S., Carter, M.: The design of a scalable, fixed-time computer benchmark. J. Parallel Distrib. Comput. **12**(4), 388–401 (1991). https://doi.org/10.1016/0743-7315(91)90008-W
26. Bachmann, P.: Die Analytische Zahlentheorie. B. G. Teubner, Leipzig, Germany (1894)
27. Bimurat, Z., Kim, Y., Ismailova, R., Sagindykov, B.: Methods of navigating algorithmic complexity: big-oh and small-oh notations. Sci. J. Astana IT Univ., 160–181, Astana (2023). https://doi.org/10.37943/15DNLB5877
28. Mala, F.A., Ali, R.: The Big-O of mathematics and computer science. J. Appl. Math. Comput. **6**(1), 1–3 (2022). https://doi.org/10.26855/jamc.2022.03.001
29. Bae, S.: Big-O notation. In: JavaScript Data Structures and Algorithms, pp. 1–11. Apress, Berkeley, California (2019). https://doi.org/10.1007/978-1-4842-3988-9_1

30. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2009)
31. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (2018)
32. Emdad, M.R.B.: Deep learning is the core method of machine learning. Int. J. Eng. Res. Technol. **6**(12), 51–55 (2017)
33. Han, Z., Hong, M., Wang, D.: Signal Processing and Networking for Big Data Applications. Cambridge University Press, Cambridge (2017).https://doi.org/10.1017/9781316408032
34. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J.: An introduction to deep reinforcement learning. Found. Trends® Mach. Learn. **11**(3–4), 219–354 (2018). https://doi.org/10.1561/2200000071
35. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. CoRR, pp. 1–14 (2013). https://doi.org/10.48550/ARXIV.1312.6114
36. Reitman, W.: Heuristic decision procedures, open constraints, and the structure of Ill-Defined problems. In: Human Judgments and Optimality, pp. 282–315. John Wiley & Sons, New-York (1964)
37. Claeys, D.: Architecture & complexité : Un modèle systémique du processus de (co)conception qui vise l'architecture, Doctoral thesis, UCLouvain, Louvain-la-Neuve, Belgium (2013)
38. Oladipupo, T.: Types of machine learning algorithms. In: Zhang, Y. (ed.) New Advances in Machine Learning. InTech, pp. 19–51 (2010). https://doi.org/10.5772/9385
39. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
40. Murphy, K.P.: Probabilistic Machine Learning: An Introduction. The MIT Press, Cambridge (2022)
41. Osisanwo, F., Awodele, O., Hinmikaiye, J., Olakanmi, O., Akinjobi, A.J.: Supervised machine learning algorithms: classification and comparison. Int. J. Comput. Trends Technol. **48**(3) pp. 128–138 (2017). https://doi.org/10.14445/22312803/IJCTT-V48P126
42. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006). https://doi.org/10.1126/science.1127647
43. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008)
44. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
45. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In : Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, California, vol. 1, no. 5 pp. 281–297 (1967)
46. Feller, W.: An Introduction to Probability Theory and Its Applications. Wiley, New York (1976)
47. Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to Linear Regression Analysis. John Wiley & Sons (2012)
48. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge (2016)
49. James, G., Witten, D., Hastie, T., Tibshirani, R. (eds.): An Introduction to Statistical Learning: With Applications in R. Springer, New York (2013)
50. Chaillou, S.: AI + Architecture | Towards a New Approach. Harvard University (2019)
51. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014)
52. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929 (2016). https://doi.org/10.1109/CVPR.2016.319

53. Camacho, E.F., Bordons, C.: Simple implementation of GPC for industrial processes. In: Model Predictive Control, pp. 81–126. Springer, London (2007). https://doi.org/10.1007/978-0-85729-398-5_5
54. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)? – arguments against avoiding RMSE in the literature. Geosci. Model Dev. **7**(3), 1247–1250 (2014). https://doi.org/10.5194/gmd-7-1247-2014
55. Stehman, S.V.: Selecting and interpreting measures of thematic classification accuracy. Remote Sens. Environ. **62**(1), 77–89 (1997). https://doi.org/10.1016/S0034-4257(97)00083-7
56. Louëdec, J., Chevalier, M., Garivier, A., Mothe, J.: Algorithmes de bandits pour la recommandation à tirages multiples. Document numérique **18**(2–3), 59–79 (2015). https://doi.org/10.3166/dn.18.2-3.59-79
57. Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. Biometrika **70**(1), 41–55 (1983). https://doi.org/10.1093/biomet/70.1.41
58. von Bertalanffy, L.: General System Theory: Foundations Development. George Braziller, New York (1968)
59. Miller, J.G.: Living Systems. McGraw-Hill, New York (1978)
60. Le Moigne, J.-L.: La modélisation des systèmes complexes. Dunod, Malakoff (1990)
61. Forrester, J.W.: Industrial Dynamics. MIT Press, Cambridge (1961)
62. Morin, E.: Introduction à la pensée complexe. Ed. du Seuil, Paris (1990)
63. Dorst, K.: The core of 'design thinking' and its application. Des. Stud. **32**(6), 521–532 (2011). https://doi.org/10.1016/j.destud.2011.07.006
64. Lawson, B.: How Designers Think: The Design Process Demystified. Elsevier Architectural Press, Oxford (2006)
65. Schön, D.A.: The Reflective Practitioner: How Professionals Think in Action. Basic Books, New York (1983)
66. Rabardel, P.: Les hommes et les technologies: approche cognitive des instruments contemporains. Colin, Paris (1995)
67. Rabardel, P.: From artefact to instrument. Interact. Comput. **15**(5), 641–645 (2003). https://doi.org/10.1016/S0953-5438(03)00056-0
68. Oxman, R.: Think-maps: teaching design thinking in design education. Des. Stud. **25**(1), 63–91 (2004). https://doi.org/10.1016/S0142-694X(03)00033-4
69. Castro Pena, M.L., Carballal, A., Rodríguez-Fernández, N., Santos, I., Romero, J.: Artificial intelligence applied to conceptual design. a review of its use in architecture. Autom. Constr. **124**, 103–115 (2021). https://doi.org/10.1016/j.autcon.2021.103550
70. Clark, A., Charlmers, D.: The Extended Mind, vol. 58, no. 1, pp. 7–13. University Press, Oxford (1998)
71. Stiegler, B.: La technique et le temps 1 La faute d'Épiméthée. Galilée, Paris (1994)
72. Latour, B.: La science en action: introduction à la sociologie des sciences. la Découverte, Paris (2005)
73. Yaneva, A.: Latour for Architects: Thinkers for Architects. Routledge, London (2022). https://doi.org/10.4324/9780429328510